Amendments to the Claims

1. (Currently Amended)   A simulator of ~~intelligent~~ client workstations, ~~at level 2 of an OSI model for generating one or more unique network data frames for transmission, the simulator~~ comprising:

at least one ~~or more~~ protocol application ~~modules~~ module for generating ~~one or more~~ data streams that emulate ~~one or more~~ client requests from at least one application running at the ~~one or more~~ client workstations;

~~one or more protocol stacks each representing a group of protocols at multiple layers, said one or more protocol stacks coupled to said one or more~~ at least one protocol stack component associated with the at least one protocol application module ~~modules~~ for ~~and receiving said one or more data streams to~~ encapsulating ~~encapsulate said~~ the ~~one or more~~ data streams according to at least one ~~with associated one or more~~ protocol stack by providing the data streams with header data according to the at least one protocol stack ~~stacks,~~ thereby providing encapsulated data streams; and

a data delivery module associated with the at least one protocol stack component for ~~receiving one or more encapsulated data streams and~~ inserting a unique hardware address identifier into each of the ~~said one or more~~ encapsulated data streams to generate ~~one or more~~ network data frames for delivery to a system under test;

wherein, each ~~inserted~~ of the unique hardware address ~~identifier~~ identifiers identifies an associated one of the ~~corresponding to an emulated~~ client ~~workstation~~ workstations ~~that originated the client request associated with said one or more encapsulated data streams~~.

2. (Cancelled)

3. (Currently Amended) The simulator of ~~intelligent workstations at level 2 as claimed in~~ claim 1, ~~the simulator~~ further comprising ~~including~~:

a scripting interface module associated with ~~coupled to said~~ the at least one ~~or more~~ protocol application module ~~modules~~, the scripting interface module receiving ~~one or more~~ emulated user actions to interpret, and passing ~~pass said~~ the emulated user actions to ~~said~~ the at least one ~~or more~~ protocol application module ~~modules~~ for use in generating ~~said~~ the ~~one or more~~ data streams thereof.

4. (Currently Amended) The simulator of ~~intelligent workstations at level 2 as claimed in~~ claim 3, wherein the scripting interface module comprises ~~includes~~:

at least one ~~or more~~ script ~~scripts~~ having ~~one or more~~ instructions for emulating the emulated user actions~~, wherein said emulated user actions are received from the script~~.

5. (Currently Amended) The simulator of ~~intelligent workstations at level 2 as claimed in~~ claim 3, wherein the scripting interface module further comprises ~~includes~~:

a service module that interprets ~~said~~ the ~~one or more~~ instructions for emulating the emulated user actions.

6. (Currently Amended) The simulator of ~~intelligent workstations at level 2 as claimed in~~ claim 3, ~~wherein the simulator~~ further comprising ~~includes~~:

a command and control service module associated with ~~coupled to said~~ the scripting interface module for enabling dynamic loading and execution of ~~said~~ the ~~one or more~~ instructions emulating the user actions.

7. (Currently Amended) The simulator of ~~intelligent workstations at level 2 as claimed in~~ claim 1, ~~the simulator~~ further comprising ~~including~~:

a command and control service module associated with ~~coupled to said~~ the at least one ~~or more~~ protocol stack component ~~stacks~~ for enabling dynamic loading of ~~said~~ the at least one ~~or more~~ protocol stack ~~stacks~~.

8.   (Currently Amended)   The simulator of ~~intelligent workstations at level 2 as claimed in~~ claim 1, ~~the simulator~~ further comprising ~~including~~:

an application programming interface ~~interposed between said one or more protocol stacks and the data delivery module, said application programming interface~~ for enabling ~~each one of said~~ the at least one ~~or more~~ protocol stack component ~~stacks~~ to communicate with the data delivery module ~~without modifications to any one of said one or more protocol stacks and the data delivery module~~.

9.   (Currently Amended)   The simulator of ~~intelligent workstations at level 2 as claimed in~~ claim 1, ~~the simulator~~ further comprising ~~including~~:

an application programming interface ~~interposed between said one or more protocol application modules and said one or more protocol stacks, said application programming interface~~ for enabling the ~~each~~ at least one ~~of said one or more~~ protocol application module ~~modules~~ to communicate with ~~each one of said~~ the at least one ~~or more~~ protocol stack component ~~stacks without modifications to any one of said one or more protocol application modules and said one or more protocol stacks~~.

10.   (Currently Amended)   The simulator of ~~intelligent workstations at level 2 as claimed in~~ claim 3, ~~the simulator~~ further comprising ~~including~~:

an application programming interface ~~interposed between the scripting interface module and said one or more protocol application modules, said application programming interface~~ for enabling the scripting interface module to communicate with ~~each one of said~~ the at least one ~~or more~~ protocol application module ~~modules without modifications to any one of said one or more protocol application modules and the scripting interface module~~.

11.   (Currently Amended)   The simulator of ~~intelligent workstations at level 2 as claimed in~~ claim 1, wherein ~~said~~ the at least one ~~or more~~ protocol stack ~~stacks~~ is associated with at least one of ~~a protocol suite selected from the group comprising~~:

a TCP/IP protocol ~~suite~~;

a UDP/IP protocol ~~suite~~; and

an SSL <u>protocol</u> ~~suite~~.

12. (Cancelled)

13. (Cancelled)

14. (Currently Amended) The simulator of ~~intelligent workstations at level 2 as claimed in~~ claim 1, <u>further comprising:</u>

~~wherein said network data frames are transmitted to a system under test, said simulator is further enabled to receive~~ <u>means for receiving</u> ~~server~~ <u>and checking</u> responses to ~~said~~ <u>the</u> client requests <u>from</u> ~~and check said~~ <u>the</u> system under test ~~responses~~ to ensure correct <u>operation of the</u> system under test ~~operation~~.

15. (Currently Amended) The simulator of ~~intelligent workstations at level 2 as claimed in~~ claim 1, ~~wherein said simulator is enabled to~~ <u>further comprising:</u>

<u>means for capturing</u> ~~capture~~ and <u>maintaining</u> ~~maintain~~ client state data from <u>the</u> ~~a~~ system under test ~~and facilitate client transmissions using said state data~~.

16. (Currently Amended) A method for <u>simulating</u> ~~generating network data frames for simulation of a plurality of intelligent~~ <u>client</u> workstations ~~at level 2 of a protocol stack, the method~~ comprising:

generating ~~one or more~~ data streams representing ~~one or more~~ emulated client requests from <u>at least one application running on the client</u> ~~one or more~~ workstations ~~identifier~~;

inserting header data associated with <u>at least one</u> ~~a selected~~ protocol stack into ~~said~~ <u>the</u> ~~one or more~~ data streams to generate ~~one or more~~ protocol encapsulated data frames; and

adding ~~the~~ unique <u>hardware</u> address <u>identifiers</u> ~~identifier~~ to ~~said~~ <u>the</u> ~~one or more~~ protocol encapsulated data frames to generate network data frames for transmission to a system under test<u>;</u>

wherein each of the unique hardware address identifiers identifies an associated one of the client workstations.

17.   (Currently Amended)   The method ~~for generating network data frames as claimed in~~ of claim 16, ~~the method~~ further comprising ~~including~~:

receiving user actions for initiating emulation of ~~said~~ the emulated client requests.

18.   (Cancelled)

19.   (Cancelled)

20.   (Currently Amended)   The method ~~for generating a generalized network data frames as claimed in~~ of claim 16, ~~the method~~ further comprising ~~including~~:

dynamically loading ~~said~~ the at least one ~~selected~~ protocol stack for generating ~~said~~ the ~~one or more~~ protocol encapsulated data frames.

21.   (Currently Amended)   The method ~~for generating a network data frames as claimed in~~ of claim 16, ~~the method~~ further comprising ~~including~~:

receiving and checking responses associated with ~~said~~ the emulated client requests from the system under test ~~and checking said responses~~ to ensure correct operation of the system under test.

22.   (Currently Amended)   The method ~~for generating a network data frames as claimed in~~ of claim 16, ~~the method~~ further comprising ~~including~~:

capturing and maintaining client state data from the system under test ~~and facilitating transmissions using said state data~~.

23.   (Currently Amended)   The method ~~for generating a network data frames as claimed in~~ of claim 17, ~~the method~~ further comprising ~~including~~:

formulating ~~said~~ the user actions into ~~a plurality of~~ transactions for communication to and from the system ~~a unit~~ under test, wherein ~~said~~ the ~~one or more~~ data streams ~~representing emulated client requests~~ are generated based on ~~from said~~ the ~~plurality of~~ transactions.

24. (Currently Amended) A program storage device readable by machine, tangibly embodying a program of instructions executable by the machine to perform a ~~the~~ method ~~steps of generating network data frames~~ for simulating ~~simulation of a plurality of~~ client ~~intelligent~~ workstations ~~at level 2 of a protocol stack~~, the method ~~steps~~ comprising:

generating ~~one or more~~ data streams representing ~~one or more~~ emulated client requests from at least one application running on the client ~~one or more~~ workstations;

inserting header data associated with at least one ~~a selected~~ protocol stack into ~~said~~ the ~~one or more~~ data streams to generate ~~one or more~~ protocol encapsulated data frames; and

adding ~~a~~ unique hardware address identifiers ~~identifier~~ to ~~said~~ the ~~one or more~~ protocol encapsulated data frames to generate network data frames for transmission to a system under test;

wherein ~~,~~ each of the ~~inserted~~ unique hardware address identifiers ~~identifier~~ identifies an associated one of the ~~corresponding to an emulated~~ client workstations ~~workstation that originated the client request associated with said one or more encapsulated data frames~~.

25. (Currently Amended) The program storage device as claimed in claim 24, wherein the method ~~steps~~ further comprises ~~comprising~~:

receiving user actions for initiating emulation of ~~said~~ the emulated client requests.

26. (Cancelled)

27. (New) The simulator of claim 1, wherein:

the unique hardware address identifiers comprise media access control (MAC) identifiers.

28.   (New)   The simulator of claim 1, wherein:

the at least one protocol application module generates the data streams that emulate the client requests from a plurality of applications running at the client workstations.

29.   (New)   The simulator of claim 28, wherein:

the at least one protocol stack component encapsulates the data streams thereof according to different protocol stacks by providing different ones of the data streams with header data according to different ones of the different protocol stacks, thereby providing the encapsulated data streams.

30.   (New)   The simulator of claim 1, wherein:

the at least one protocol stack component encapsulates the data streams thereof according to different protocol stacks by providing different ones of the data streams with header data according to different ones of the different protocol stacks, thereby providing the encapsulated data streams.

31.   (New)   The method of claim 16, wherein:

the unique hardware address identifiers comprise media access control (MAC) identifiers.

32.   (New)   The program storage device of claim 24, wherein:

the unique hardware address identifiers comprise media access control (MAC) identifiers.